University of Tripoli **Faculty of Engineering** **Electrical and Electronic Engineering**
EE334 **First Exam.** **Time: 75 min**

==========

Q1) a.Write a subroutine that will convert all lower case characters in a string to upper case. A lower case ASCII character has a value between 61H ('a') and 7Ah ('z'). To convert to lower case, subtract 20H or clear bit B5 to a zero. The starting string address is passed in register BX, and the string is NULL TERMINATED (last byte is 00h). Other characters in the string that are not lower case characters ';'; should be unaffected.

```
DEC BX
again :
        INC   BX
        MOV   AL   , [BX]
        CMP    AL   ,  00H
        JZ      finish
        CMP   AL    ,  61H
        JB      again
        CMP   AL    , 7AH
        JA     again
       SUB    AL    , 20H    ; or  AND AL , 11011111b
       MOV   [BX]   ,  AL
       JMP    again
Finish                                                                    :
```

b.Why you use Memory Segmentation in 8086 Microprocessor?

Memory segmentation in the 8086/8088 is used to allow the processor to access more than 64kb of memory, even though it is only a 16-bit processor. Each segment register allows access to one of 64k 64kb segments, each overlapping by 16 bytes, with the total addressibility being 1MB.

c.If ES = D321H, then the range of physical addresses for the extra segment is:? ……

D3210h to E320F

Q2)a. What is the largest signed integer that may be stored in 32 bits?

$2^{31-1}-1$

b.What will be the values of the Sign, and Zero flags after the following instructions have executed?

```
mov ax,620h        sub ah,0F6h
```

SF= 0 , ZF=0 => result 10h

c. If we declare the three arrays AW, BW, and CW of words by

```
AW  DW 000Ah, 010Ah, 020Ah, 030Ah, 040Ah
BW  DW 000Bh, 010Bh, 020Bh, 030Bh
CW  DW 000Ch, 010Ch, 020Ch, 030Ch, 040Ch, 050Ch
```

Fill in the contents of the specified registers in the following code as hex-digit numbers:

```
Mov ax,[BW + 2] ; ax =  010Bh
Mov ax,[AW + 14] ;ax = 020Bh
Mov ax,[BW–4] ;   ax = 030Ah
Mov ax,1234h
 Xchg ah,al ;          ax = 3412H
```

```
MOV BX,B372h
MOV AX,BX ;      AX=B372H
MOV BX,B372h
MOV DX,BL ;      DX=ERRORE
```

Q3) a. Trace the following code:

```
.data
    FinalResult DWORD 11223344h
.code
    MOV AL, 3
    MOV BL, 2
    LEA SI, FinalResult    ; 3rd line????????
    MOV CX, 4
```

```
L1:
    MOV BYTE PTR [SI], AL
    SUB BYTE PTR [SI], BL
    MOV AL, BL
    MOV BL, BYTE PTR [SI]
    INC SI
    LOOP L1
```

1) What is the value stored in FinalResult after the execution of the above code?
   01010001h
2) Suppose the 3rdline in the code is changed to MOV SI,0 so as to initialize SI to 0.Change the code within the loop to produce the same results

```
L1:
    MOV BYTE PTR FinalResult[SI], AL
    SUB BYTE PTR FinalResult[SI], BL
    MOV AL, BL
    MOV BL, BYTE PTR FinalResult[SI]
    INC SI
    LOOP L1
```

b. trace the following procedure ,

1- what does the procedure do ? CONVERT FORM DECIMAL TO BINARY

2- what is the properties of STR1 and STR2 and used for what?

A BYTE IS THE SIZE OF EACH ELEMNT , STR1 USED TO STORE THE REMINDER OF THE DEVISION (0 OR 1) , while STR2 will store same value of STR2 but reverse order.

3 – Explain the function of each instruction used and why ?

1- get the offset of str1
2- get the offset str2
3- make ax =00 ie, ah =00 for division AL/02 8bit divide by 8 bit
4- put the decimal number in al to divide it by 2 (decimal to binary)
5- division operation
6 - convert the reminder of the division to ASCII by adding it 30h
7 - store the reminder (ASCII format) in SI (STR1)
8- increment SI to point to next element in STR1
9- save the result of the division into NUM to do next division of next cycle
10- check the value of AL , i.e convert operation finished or not
11 - if AL greater than zero , go on for next division . i.e go to third instruction

---

now to get the binary in correct order we should reverse STR1 and store it in STR2
12 - if the division finished , decrement SI by one cause it pointing to last value of STR1 +1 .
13- read element in STR1 (SI pointing to it) and store it in BL
14 store BL (element of STR1) at STR2 (DI pointing to it)
15- and 16- decrement SI an increment DI ,
17- and 18 condition for ending loop two by comparing SI with 0 or the offset of STR1

```
    LEA SI,STR1
   LEA DI,STR2
L1: MOV AX,00
   MOV AL,NUM    ; NUM is variable has decimal value
   DIV 2
   ADD AH,30H
   MOV BYTE PTR[SI],AH
   INC SI
   MOV NUM,AL
```

```
   CMP AL,0
   JG L1
      DEC SI
L2:MOV BL,BYTE PTR[SI]
   MOV BYTE PTR [DI], BL
   DEC SI
   INC DI
   CMP SI,0
   JNE L2
```

********** **********GOOOOOOOOOOOOD LUCK **********